REMARKS

Rejection under 35 USC section 112 paragraph 2

This rejection is respectfully traversed. Applicants respectfully submit that the Examiner is asking Applicants to narrow their claims under the guise of an indefiniteness rejection. The art does not justify narrowing the claims. Applicants are entitled to draft their claims sufficiently broadly to cover more than one embodiment.

Art rejections

The art rejections are respectfully traversed.

Since the references are many and/or complex, Applicants will confine their remarks to those portions of the references cited by the Examiner, except as otherwise indicated. Applicants make no representation as to the contents of other portions of the references.

The prior comments are incorporated by reference and supplemented as follows:

Claims 1, 13, 25, and 49, and 76-79

As pointed out before, the specification contains the following definitional section at the bottom of page 6:

> Herein, the following definitions will be used. A data source or document "schema" describes the structures and types for data or documents. An "annotation" furnishes the schema with mapping functions that connect data from heterogeneous data sources and target data segments.

Applicants respectfully submit that the Examiner continues to ignore Applicants' definition of schema and instead inserts his own, namely "is an annotated schema in that it maps retrieved data into documents that conform to the standard." The Examiner is not at liberty to substitute his definition for that set forth by Applicants in their specification. Applicants are allowed to be their own lexicographers. In order to be a schema, an item must describe structures or types for data or documents. An annotation must furnish the schema with mapping functions. The Examiner has failed to indicate where either the XEDI or the EXTOL reference teaches or suggests an annotated schema as defined by Applicants.

Indeed, Applicants understand these references to do the following:

1. First, a line-by-line translation of EDI documents into simplified XML, without using any schema, but rather a dictionary per XEDI p. 13, 2nd paragraph. A dictionary is not a schema. It does not describe type or structure of data. It just translates one thing into another.
2. Second, translation of the simplified XML into more complete XML or HTML using style sheets, i.e. using XML Style Sheet Language Transformations ("XSLT"), per Xedi, page 13 2nd par. and EXTOL page 19, second full paragraph. Style Sheets are programmed using prior knowledge of the source and target formats. They are thus mapping functions that connect data sources and targets. They may have some information about source structures and types, but they cannot be considered annotated schema, because no annotation furnishes anything to a schema. In order for there to be an annotated schema, there must first be a schema and then an added annotation.

Enclosed is a copy of Zhou et al., "TREX: DTD-Conforming XML to XML Transformations" SIGMOD 2003. This document is not prior art, but it does explain current knowledge about the techniques that were used in the references. This document explains that XSLT translations cannot be guaranteed to conform to DTD's. Since DTD's are the type of schema relevant to the Zhou article, this means that XSLTconversions are not guaranteed to be schema-conforming. XSLT conversions have to be checked. Since the claimed invention uses schema for retrieval, it is schema-conforming, without additional checking. In other words, the output document will have a desired format. Accordingly, the invention has a functional advantage over the references.

In the response to arguments section, the Examiner states that Fig. 9 of XEDI shows an annotated schema. Applicants respectfully disagree. This diagram shows use of a data dictionary for the first translation and transformation instruction sets for the second transformation. The box labeled "XEDI" is characterized as a translator. As explained above, neither is an annotated schema, i.e. a description of type or structure, with annotations that include mapping functions.

The Examiner has accordingly failed to present a *prima facie* case against these claims.

New claims 76 through 79 more clearly specify the nature of schema and annotations and therefore distinguish even more clearly over the references.

Claim 37

As explained in the prior response, in rejecting this claim, the Examiner reads the EDI messages as annotations to a schema. Applicants respectfully submit that the Examiner misconstrues the reference. The EDI messages are not annotations that are adapted to guide retrieval in accordance with a universal schema, unlike Applicants recited invention. As far as Applicants can tell, the EDI messages are what is to be translated or converted, or at best they are some type of explanation.

Moreover, Applicant does not find that the Examiner has responded to this argument. Applicant respectfully submits that this is improper.

Claims 42, 53, and 64

Claim 42 recites, *inter alia*, an annotated schema. Against this recitation the Examiner cites pages 12-13 of Xedi reference. Applicants have read this section and see it as setting forth problems relating to XML and EDI and also as listing organizations that are dealing with these problems. Applicants see absolutely nothing here that would teach or suggest an annotated schema, as defined by Applicants, or indeed much of any solution to anything.

Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against claim 42.

Claims 53 and 64 include recitations analogous to those discussed above with respect to claim 42.

Claim 75

Claim 75 recites using a universal annotated schema in both the conversions, i.e. the conversions into both the second and first data formats. The references are similarly deficient in failing to teach or suggest that a universal schema could be used in both conversions.

Claims 4, 8, 16, 20, 28, and 32

Claim 4 recites the same type of software engine being applied first to the at least one type of data source and then to the first electronic format. The Examiner purports first to find this in Xedi's Fig. 8. Applicants respectfully submit that the Examiner misconstrues Fig. 8. This figure shows only some servers and other equipment hooked to the Internet. No software is shown at all. Fig. 9 is clearer, since it shows that an entirely different software engine is used for the first and second translations, one being a translator and the other being a parser.

Claims 8, 16, 20, 28, and 32 are analogous with respect to this point.

Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against these claims.

Claim 6, 18, 30, 43

Here the Examiner alleges baldly that the EDI-XML translator acts as an annotated DTD. A translator is not an annotated DTD. An annotated DTD can be used by a translator, but it is not a translator itself. A DTD is a type of schema, well defined in the standards, that is used by software in creating XML documents. The Examiner cannot assert as Humpty Dumpty does in Lewis Carol's Alice through the Looking Glass that he gets to make words mean whatever he choses. Words mean what they mean. A translator is not a DTD.

Claims 18 and 30 are analogous to claim 6 with respect to the discussion in the previous paragraph.

Claims 7, 19, and 31

Claim 7 recites, *inter alia,* that the second annotated schema comprises annotations for retrieving specifications.

The Examiner purports to find annotations on pages 11-13 and Fig. 9 of XEDI. Applicants have searched these sections in vain for anything as detailed as an annotation to a schema. These seem to be high level discussions of the functions of the software, without any such detailed considerations. What language, exactly, does the Examiner believe could be an annotation?

Claims 19 and 31 include recitations analogous to those discussed above with respect to claim 7.

Claim 40

Claim 40 depends from 37 and 39 recites a universal DTD with recursive annotations. It thus relates to the field of a universal DTD.

The Examiner cites the file labeled "XEDI" in Fig. 9 of the Xedi reference as being an annotated DTD. Applicants respectfully submit that the Examiner mischaracterizes the reference. Applicants understand the item marked "XEDI" in Fig. 9 to be a translator module, not to be a DTD.

Moreover, the Examiner mischaracterizes Applicants prior argument. Applicants never admitted that the annotations in the prior art could be recursive, only that DTD's could be recursive. The DTD's are a type of schema, which describe the type and structure of XML documents. The annotations, per Applicants' definition, furnish the DTD with a mapping. Recursive DTD's fail to teach or suggest recursive annotations.

Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against claim 40.

Claim 41

The Examiner states that the alleged universal DTD of Xedi attaches unique labels to corresponding intermediate XML documents or value pairs. For this proposition, he cites Fig. 9,

& Page 12-13 of the Xedi reference. Applicants have reviewed these portions of the reference and find nothing about labels or value pairs. This is a high level reference that simply does not go into this degree of detail. Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against claim 41.

Claims 12, 24, and 36

The Examiner rejects these references over 4 references. Applicants respectfully submit that this is clearly a hindsight reconstruction using Applicants' claims as a roadmap[1] and is therefore improper. The suggestion to combine references must come from the references, not from Applicants' claims.

Moreover, Applicants respectfully submit that the Examiner mischaracterizes the Kotok reference here. Applicants find that Kotok talks about repeating a date, but no teaching or suggestion that a single annotation is repeated relating to this repetition. Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against ths group of claims.

[1] The CAFC has said

> The "as a whole" instruction in title 35 prevents evaluation of the invention part by part. Without this important requirement, an obviousness assessment might break an invention into its component parts (A + B + C), then find a prior art reference containing A, another containing B, and another containing C, and on that basis alone declare the invention obvious. This form of hindsight reasoning, using the invention as a roadmap to find its prior art components, would discount the value of combining various existing features way to achieve a new result - often the very definition of invention. Ruiz v. A. B. Chance Co., http://www.ll.georgetown.edu/federal/judicial/fed/opinions/03opinions/03-1333.html at p. 7, 357 F. 3d 1270, 2004 US App. Lexis 1325, 69 U.S.P.Q. 2d (BNA) 1686 (Fed. Cir 2004)

Claims 45-47, 56-58, and 67-69

Claim 45, 56, and 67 recite determining whether the annotated schema is reversible. As explained before, in the context of Applicants' specification, pp. 22-23, 'reversibility' means that the same annotated schema can be used for retrieval or deposit. This recitation does not mean merely that a reverse transformation exists, as the Examiner seems to think. The recitation means a determination of whether the annotated schema itself is usable in the reverse transformation.

The Examiner cites references that talk about reverse transformations, but not "reversibility" as used in Applicants' specification, pp. 22-23. The Examiner is referred to the recently decided case of Philips v. AWH Corp (Fed. Cir. Docket # 03-1269,1286 July 12, 2005), which emphasizes the importance of the specification in interpreting terminology.

The Examiner cites EXTOL relating to reversibility. However, Applicants are unable to find that EXTOL teaches or suggests that an annotated schema could be reversible, in other words that the same annotated schema could be used for retrieval and deposit. Quite on the contrary, Applicants believe, based on their own knowledge of style sheets such as are used in EXTOL, that -- not only are they unreliable, because they fail to make XML documents conform to DTD's -- but also that they are completely non-reversible. In other words, a completely different style sheet must be used for the reverse transformation than what is used in the initial transformation. Accordingly, Applicants do not see any teaching or suggestion here of any determination of reversibility.

Abjanic converts messages between AXML and BXML and reverses the transformation. However, Applicants find no teaching or suggestion, at least in the cited portions, that an annotated schema, used in such a conversion, could be reversible or could be determined to be reversible. As far as Applicants can tell, Abjanic is similarly deficient with Extol in requiring a completely separate sort of transformation for reversal.

These references in fact completely confirm the remarkable nature of Applicants' invention.

Claims 80-82 are added to emphasize that a positive determination -- i.e. a determination that an annotated schema can be used for retrieval and deposit -- is possible, which is not taught or suggested in the references.

Claims 46, 57, and 68 recite that if an annotated schema is not reversible, a revised annotated schema is created that is reversible. The Examiner purports to find this in paragraphs 85-86 and 98 of Abjanic. The Examiner finds portions of Abjanic which determine whether reversing is necessary. Determining that reversing is necessary only begs the question of how that reversing will be accomplished. Determining whether reversing is necessary fails to teach or suggest a determination of whether an annotated schema is itself reversible as that term is used in Applicants' specification.

The Examiner states that pages 18-19 of EXTOL disclose creating a reversible annotated schema. Applicants respectfully submit that the Examiner mischaracterizes the reference. There is no teaching or suggestion here that any style sheets could be reversible.

Applicants accordingly respectfully submit that the Examiner has not made a *prima facie* case against these claims.

The term reversible is to be interpreted the same way with respect to other claims, such as 47, 58, 69, 51, 62, and 73.

Claims 50, 61, and 72

In addition to recitations relating to annotated schema -- the arguments against which are defective as discussed above with respect to similar limitations -- these claims recite multiple relational databases.

Against this recitation, the Examiner again cites p. 3, par. 1 of Kotok. This paragraph apparently discusses only a single standard known as X12. Applicants find no teaching or suggestion of multiple relational databases. Applicants are at a total loss to understand why the Examiner even cites this paragraph.

Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against these claims.

With respect to most of the rest of the rejections, the arguments presented above apply here *mutatis mutandi.*

The Examiner's other rejections and/or points of argument not addressed herein would appear to be moot in view of the foregoing. Nevertheless, Applicants reserve the right to respond to those rejections and arguments at a later date. No arguments are waived and none of the Examiner's statements are conceded.

Applicants respectfully submit that they have demonstrated the claims to have been rejected in error. Allowance is accordingly respectfully requested.

Please charge any fees other than the issue fee to deposit account 50-0510. Please credit any overpayments to the same account.

Respectfully submitted by,

Anne E. Barschall, Reg. No. 31,089
914-332-1019; fax 914-332-7719
9/2/2005

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited this date with the United States Postal Service as first class mail in an envelope addressed to

Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

On 9/2/05/ (date)
By ________ (signature)

TREX: DTD-Conforming XML to XML Transformations

Aoying Zhou, Qing Wang, Zhimao Guo, Xueqing Gong, Shihui Zheng
Hongwei Wu, Jianchang Xiao, Kun Yue
Fudan University, China
{ayzhou,qingwang,zmguo,gongxq,shzheng0,hwwu,jcxiao,kuny}@fudan.edu.cn

Wenfei Fan
Bell Laboratories, USA
wenfei@research.bell-labs.com

1. Overview

There have been increasing demands for a system to support *DTD-conforming XML to XML transformations:* given any target DTD D and a source XML document S, extract data from S and construct a target XML document T such that T conforms to the predefined D. The need for this is evident in, *e.g.*, data exchange, security views, and data integration. Popular XML query languages (*e.g.*,XQuery, XSLT) cannot guarantee DTD conformance in XML to XML transformations. Type inference and (static) checking are too expensive to be used in practice; worse still, they provide no guidance for how to ensure DTD-conformance.

In response to the need we propose *TREX* (TRansformation Engine for XML), a middleware system for DTD-conforming XML to XML transformations. TREX is based on the novel notion of *XTG* (XML Transformation Grammar), which extends a DTD by incorporating XML queries into element type definitions. This allows one to specify how to extract relevant data from a source XML document via the queries, and to construct a target XML document directed by the embedded DTD. TREX efficiently evaluates XTGs by implementing several optimization techniques. XTGs and TREX provide the first systematic method and practical system to support DTD-conforming XML transformations.

2. Features of TREX

XTG and TREX have a number of salient features (see [1] for detailed discussions).

DTD-conformant specifications. With XTGs one can easily specify XML to XML transformations. The queries associated with an element type definition (DTD grammar rule) find relevant source data to generate the children of a target element, and the grammar rule is enforced when the children are created to guarantee DTD-conformance.

Data-driven semantics. XTGs are capable of handling complex DTDs. During the generation of a target document, the decisions on the choice of a *nondeterministic* (disjuncion) grammar rule and on the expansion of the target XML tree in the *recursive* case are based on the source data.

Batch and lazy modes. TREX is quite flexible by offering two evaluation modes. In the *batch* mode, it generates a complete XML document. In the *lazy* mode, it constructs a partial XML (DOM) tree, interacts with users, and expands the tree based on users' requests and interests.

Query composition, tuning and caching. TREX employs several optimization techniques: composing related XML queries, simplifying XPath expressions, and caching query results to avoid unnecessary recomputation. With these TREX efficiently support XTGs.

Although TREX is similar in spirit to the XML publishing system for relational data reported in [2], it conquers new challenges in the context of XML to XML transformations, at both the conceptual level and the implementation level.

3. System Demonstration

A prototype of TREX has been implemented in Java, using Kweelt [4] as the underlying engine for XML queries. We choose Quilt [3] to express XML queries in XTGs rather than XQuery/XSLT because we could access the source code of Kweelt to incorporate our optimization algorithms. With the prototype, the demonstration is to show how to transform DBLP publication data to an XML document that conforms to a predefined recursive and nondeterministic DTD, in both batch and lazy modes. This verifies the effectiveness of our specification and optimization techniques.

Acknowledgment: This work is supported by NSFC under Grant 60228006. Aoying Zhou is also supported by National 973 Program of China under Grant TG1998030404, and Wenfei Fan is supported in part by NSF Career Award IIS-0093168.

4. References

[1] Full version. http://www.bell-labs.com/user/wenfei/papers/sigmod03-demo.pdf.

[2] M. Benedikt, C. Y. Chan, W. Fan, R. Rastogi, S. Zheng, and A. Zhou. DTD-directed publishing with attribute translation grammars. In *VLDB*, 2002.

[3] D. Chamberlin, J. Robie, and D. Florescu. Quilt: An XML query language for heterogeneous data sources. In *WebDB*, 2000.

[4] SourceForge. Kweelt. http://kweelt.sourceforge.net.

SIGMOD 2003, June 9-12, 2003, San Diego, CA.
Copyright 2003 ACM 1-58113-634-X/03/06 ...$5.00.

YoR 9-2001-0458